

APPLICATION
FOR
UNITED STATES LETTERS PATENT

TITLE: COMPETITIVE AVAILABILITY TOOLS
APPLICANT: JEREMY WERTHEIMER AND CARL G. DEMARCKEN

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL224699106

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

July 13, 2000

Date of Deposit

Linda Wesley
Signature

Linda Wesley
Typed or Printed Name of Person Signing Certificate

COMPETITIVE AVAILABILITY TOOLS

BACKGROUND

This invention relates generally to determining airline seat availability information for use in travel planning and travel reservation systems.

Airlines institute selling policies that can change to meet supply and demand considerations to maximize profit on any given flight. When a passenger specifies an itinerary the itinerary has one or more flight segments. In order to issue a ticket for a single or multi-flight segment itinerary, each flight segment must be available. That is, each flight segment must have seats that have not been already reserved for other passengers. Availability can also be govern by whether an airline will sell to a particular passenger given characteristics of the passenger. Common characteristics which are used by airlines to decide whether or not to sell a ticket is the price that the passenger is willing to pay for the ticket, whether the passenger is using other flights on that airline, whether the passenger is a frequent flyer and so forth.

Generally, before booking a flight and issuing a ticket, the seller can send a request for availability information to the airline. In general, a request for availability is sent over a computer network to an airline and is processed in the airline's computer system. An answer to the request is provided from the system. Commonly, a message is returned to the seller. The message includes one or possibly a plurality of so-called booking codes that are labels used to designate different prices that an airline is willing to sell tickets at. Associated with these booking codes or labels are often a number of seats that the airline is willing to sell in each booking code. For example, a common booking code is the "Y"

booking code and the message may contain Y/25 meaning the Y
booking code has 25 seats. A second booking code may be the "Q"
booking code and may contain a message which says Q/0 meaning
that the Q booking code has 0 seats available. Although the
5 exact meaning of booking codes may vary from carrier to carrier,
in general most carriers will use Y booking codes corresponding
to an expensive coach class fare and a Q booking code as an
inexpensive coach class fare. The airline would make the seat at
the Y booking code available, i.e., a higher profit booking code,
10 rather than make the seat available at the Q booking code, i.e.,
a lower profit fare.

SUMMARY

According to an aspect of the invention, a competitive,
availability prediction system for predicting relative,
15 competitive availability of seating on an airline flight includes
an availability predictor that predicts seating availability on a
competitive flight, an availability system that produces an
actual availability response for a flight and decision logic that
compares the predicted answer from the availability predictor and
20 the potential answer from the availability system to establish a
decision with respect to actual availability.

According to an additional aspect of the present
invention, a method of predicting relative, competitive
availability of seating on an airline flight includes predicting
25 seating availability on a competitive flight and providing an
actual availability response for a flight. The method also
includes comparing the predicted answer from the availability
predictor and the potential answer from the availability system
to establish a decision with respect to actual availability.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a competitive airline availability system.

FIG. 2A is a flow chart of a process used in the competitive airline availability system of FIG. 3.

FIG. 2B is a flow chart of exemplary decision logic used in the competitive availability system of FIG. 3.

FIG. 3 is a block diagram of an availability database.

FIG. 4 is a block diagram of a predictor using the availability database of FIG. 4.

FIGS. 5 and 6 are flow charts of processes used with the availability database.

FIG. 7 is a block diagram of a threshold level predictor.

FIG. 8 is a block diagram of an availability table.

FIG. 9 is a block diagram of an availability predictor.

FIG. 9A is a diagram showing an exemplary query.

FIG. 10A is a block diagram of an availability predictor of FIG. 3 using an exponential modeling algorithm.

FIG. 10B is a block diagram of an availability predictor using a decision tree algorithm.

DETAILED DESCRIPTION

Referring now to FIG. 1, a competitive availability system 10 is shown. The competitive availability system includes a filter 12 that is used to filter queries received by the system 10. The filter 12 can have rules that allow the filter 12 to pass through those queries that correspond to flights supported by a user of the competitive availability system 10, as well as selected competitors of that user. Other criteria can also be used in the filter 12. The competitive availability system 10 produces a prediction of the availability of a seat on a

competitor's flight or flights and how a competitor or competitors may respond to an availability request. The user of the competitive availability system 10 can decide whether and how to adjust its response from the availability system 74.

5 In the typical case, the user of the competitive availability system 10 would be an airline that desires to modify its actual availability response to an availability query that it receives based on how it expects a competitor airlines might respond to the same query.

10 The filtered queries provided from filter 12 are fed to one or more availability predictors generally denoted as 14. The availability predictors 14 are provided for each competitor for which the user of the competitive availability system 10 desires to compare airline availability responses. Preferred techniques for implementing the availability predictor 14 are described in conjunction with FIGS. 3-11 below.

15 The filtered queries are also fed to the actual or a simulated availability system 16 of the airline that owns or uses the competitive availability system 10. The availability predictor 14 and the availability system 16 each produces answers. The availability predictor produces a predicted answer for the competitor and the availability system produces a potential availability answer for the user of the competitive availability system 10. These answers are fed to decision logic 18. The decision logic 18 compares the answers to determine whether or not the actual answer that will be provided from the user's availability system 16 should be modified to take into consideration the relative competitive situation of the competitor represented by the availability predictor 14. The decision logic 18 produces an output that corresponds to a message that is sent to the availability answer modifier logic 20. The message from the decision logic can take on a number of

forms and have a number of states. One exemplary message form can be one that biases the modifier logic 20 in one direction or another. Such a message form can have several states. One state could be a neutral state with respect to biasing an availability answer provided from the availability system. That is, a neutral answer can be returned and would not tend to modify the potential answer received from the availability system 16. A second state could bias towards answering that a seat is available. A third state could towards answering that a seat is not available. The state would depend upon the relative competitive position of the competitor represented by the availability predictor 14.

The availability answer modifier logic 20 receives the potential availability answer provided from the availability system 16 and the message from the decision logic 18 and selectively modifies the potential availability answer received from the availability system 16 in accordance with the bias of state of the message from the decision logic 18. The potential, modified availability answer is the answer that is sent to the entity that initiated the query to the airline's availability system 16.

Referring now to FIG. 2A, a process 30 incorporating high level decision logic 18' and modification logic 20, that is used in the system of FIG. 1 is shown. The process 30 receives 32 the predicted and potential, actual responses from the availability predictor 14 (FIG. 3) and the availability system 16 (FIG. 3) and compares 34 the predicted and potential, actual responses to arrive at a decision of whether to bias a modification towards making a seat more available, less available or to remain neutral. The process 30 will modify 36 the potential, actual availability response based upon the comparison and may return 38 the potential, actual availability response or a modified actual availability response to the entity that issued

the query in the first instance.

Referring now to FIG. 2B, an exemplary process flow 40 for decision logic 18 (FIG. 1) is shown. It should be understood, however, this is merely an exemplary process flow and many other process flows may alternatively be used. The process flow 40 examines the competitor's relative position to the user of the availability system 10. In particular, the process will ask 42 whether the availability predictor 14 predicts that the competitor has available seats. If the prediction is that the competitor does not have available seats, then the decision logic could return 44 a message indicating "no bias" i.e., the modification logic would not change the potential answer issued by the availability system 16. If the competitor does have available seats, the decision logic 18 can determine 46 whether the competitor's available booking codes are at a lower price than those which the availability system 14 indicates the user of the system 10 can offer. If the competitor's available booking codes are not at a lower price, then the system can test 48 whether the original query was for a "low cost fare." If the query was for a "low cost fare," the system can return 50 a bias towards making the seat "not available." Otherwise, the system can return 52 "no bias."

If the competitor's available booking codes are at a lower price than those being offered by the user of the system 10, the system can ask whether 54 the query was for a "high cost fare." If the query was for a "high cost fare," the decision logic 18 can return 56 a bias towards making the seat "available." If it was not for a high cost fare, the system can return a message of "no bias" 118.

Alternatively, the messages that are returned could be change the availability message from the availability system 16, rather than merely biasing a change in the availability message.

The modification logic 20 can change the actual availability answer from the availability system 16 using various analytical techniques. For example, the messages can have a bias that is set as a threshold value and the modification logic can
5 examine the messages to determine if the threshold value calculated in the decision logic 18 exceeds the threshold set in the modification logic 20. The threshold can be set taking into considerations factors such as current inventory for a flight, days to departure, booking code availability and so forth.

10 It should be understood that many other considerations can enter into the decision logic 18 to determine whether or not an availability answer should be modified based upon a competitor's relative competitive position. Moreover, although only a single availability predictor 14 for a single competitor is shown in FIG. 1 and described in FIGS. 2A and 2B, it should be apparent that many more availability predictors 14 can be provided, one for each competitor of the user of the system 10. The decision logic 18 and the process flow would be substantially similar except for modifications needed to compare the relative
15 competitive positions of the additional competitors.
20

Referring now to FIG. 3, a first embodiment 65a of an availability predictor 65 includes a database 70, a database engine 80 and a predictor process 90. The database 70 stores availability queries and answers, as shown. Several sources of
25 availability information can be used. One type of source is actual availability answers. However, certain considerations can make these types unattractive or not available. Other sources therefore could include public sources of seat availability such as ticket information, passenger lists, etc. Also AVS messages
30 as described below could be used. The queries and the results of these queries can be forwarded and stored in the database 70. The database 70 will contain the query such as shown below. For

a query involving a single flight:

Airl	Flt#	Orig	Dest	Date	TripOrigin	TripDest	SoldIn	SoldBy
AA	1822	BOS	DEN	25MAR99	BOS	LAX	US	Amer.Expr.

or for a query involving multiple flights:

5	Airl	Flt	Orig	Dest	Date	TripOrigin	TripDest	SoldIn	SoldBy
	AA	1822	BOS	DEN	25MAR99	BOS	LAX	US	Amer.Expr.
	DEN	LAX	25MAR99	BOS		LAX	US	Amer.Expr.	AA 0421

A result will generally comprise a message such as shown below:

Airl	Flt#	Orig	Dest	Date	BookingCodes&Counts
AA	1822	BOS	DEN	25MAR99	F0 C0 Y9 M5 K5 L0 Q0

or

Airl	Flt#	Orig	Dest	Date	BookingCodes&Counts
AA	1822	BOS	DEN	25MAR99	F0 C0 Y9 M5 K5 L0 Q0
AA	0421	DEN	LAX	25MAR99	F1 C0 Y4 M5 K1 L1 Q1

Additional information can be stored in the database 70 which may typically be generated by the availability predictor 65a. For example, the query can be stored along with an entry that corresponds to the time and/or date that the query was stored, received, and/or generated. The source of the query can also be noted. In addition, other information may also be stored with the query such as characteristics of the customer or traveler. Such characteristics may include the traveler's nationality, point of purchase or status such as whether the traveler is a frequent flyer or whether the traveler is booking other flights on the airline to which the query was directed and so forth. The database 70 can also be populated by routine

direct queries from the various sources even in the absence of queries made to the predictor so that, when a question is asked of the predictor, it is less likely that a direct query would have to be made. For example, the database 70 may be populated during off peak times for travel agents or may be simply populated with such routine queries when the system is not otherwise in use.

The database engine 80 populates the database 70. The engine 80 can produce queries of certain types depending upon the relative factors involved in any particular flight and/or airline. Such routine queries could be automatically produced by the database engine 80 for those markets and/or flights in which air travel is particularly heavy or during such periods of time where air travel between particular origins and destinations would be particularly heavy.

Referring now to FIG. 4, the predictor process 90 that uses the database 70 to provide predicted availability answers is shown. The predictor process 90 includes an update process 92 that interfaces with the query database 70 (FIG. 3) and database engine 80 to make sure that the query database 70 contains the most current information available for the availability predictor 90. The update process 92 takes responses that are received from queries made by various sources, and populates them into the query database 70 as appropriate. The predictor 90 also includes a look-up and retrieval process 94 that interfaces with the query database 70, as well as other sources of availability data. In response to a query, the look-up and retrieval process 94 produces either a prediction for the answer of the query or an actual answer depending upon whether the look-up and retrieval process retrieves an answer from the database 70 or other sources.

Referring now to FIG. 5, the update process 92 receives a query 102 from either the availability predictor 90 or from other sources, as described in conjunction with FIG. 3. The update process 92 assigns 104 a time, date, source, and user characteristic parameters, if available, as appropriate and stores 106 the query along with the answer and the assigned parameters in the query database 70.

Referring now to FIG. 6, the look-up and retrieval process 94 receives a query. Queries can include information such as airline, flight number or numbers, origin and destination airports, and travel date. In addition, the information can also include trip origin and destination if different than the origin and destination of the queried flight-segments. Queries may also include information about the selling location or agency. For travel involving multiple flight-segments, individual queries may be constructed for each flight segment, or a single query for multiple flight-segments might be constructed.

The look-up and retrieval process 94 will look up 112 the received query in the query database 70 by attempting to match the query fields such as airline, flight number/numbers, date, trip origin and destination, sale location and agency. If a stored query is found 114 in the query database 70 that matches the received query or which is substantially close in characteristics to the received query, the process 94 will retrieve 116 the stored answer. The process 94 will determine if the stored answer is stale 118 by comparing the time of the query to a threshold time that can be either a preset threshold such as a certain number of minutes, hours or days or preferably a variable threshold that is determined in accordance with a threshold level predictor 120 (FIG. 7). If the answer is not stale, then the look-up and retrieval process 94 will return 120 the stored answer as a prediction of the availability of a seat

on a particular flight according to the availability query.

If the query was not found in the database 70 or if the stored query which was found is stale, the look-up and retrieval process 94 optionally can determine 122 whether or not to use another predictor such as one of the predictors to be described in conjunction with FIGS. 8-11. If the look-up and retrieval process 94 has this option, the process 94 will return 124 the prediction from those predictors, as the prediction from the availability predictor 65a. Otherwise, if the look-up and retrieval process 94 does not have a predictor or does not trust the predictor, then the process can send 126 an actual availability query to the and availability system 66 (FIG. 2) or other sources. The answer that is received 128 from the availability system 66 or other sources is returned 130 as the answer and can be used to update 130 the database 70. The database 70 can be implemented using various approaches including hierarchial, relational or object oriented databases, or alternatively, a software or hardware cache. In addition, the answer can include a confidence factor based on whether the query is stale or whether an actual query was performed.

Referring now to FIG. 7, a threshold level predictor 140 is shown. The threshold level predictor 140 can be fed by query factors 142 such as the date of a flight, origin and destination of the flight, size of the airplane and so forth and also fed by predictor inputs 144 that determine relative weights, for example, to assign to each one of the query factors. The threshold level predictor 140 can determine a threshold time interval that can change over time. The threshold level predictor 140 can be used by the look-up and retrieval process 94 to determine whether a stored query is stale. The threshold level predictor 140 can be a mechanism that models or predicts a rate at which seats are reserved on a particular airline given

the inputs or the time that an airline adjusts parameters that affect how availability seats are distributed among various booking codes. The model can take into consideration historical rates at which flights or families of flights are sold on
5 different dates, aircraft capacity, external events such as strikes or sales and so forth.

For a very simple example, the threshold predictor 140 could be a table similar to FIG. 8 that includes for every airline/booking-code/days-before-departure entry, a number of
10 hours after which a database answer will be considered stale. This table could be trained on historical data by recording for each airline/booking-code/days-before-departure combination the average maximum number of hours prior to a query that other queries returned the same answer. For example, if in the past on American 3 days before departure in booking code Q, query answers remained the same for an average of 8 hours, then 8 hours would be stored in the table, and database queries for
AA/Q/3-days-before-departure would be considered stale if they were more than 8 hours old.

Several options are provided for returning the predictions and/or answers from the look-up and retrieval process
20 94. For example, the look-up retrieval process 94 can simply return a true/false indication indicating that a seat conforming to the parameters of the query is available or is not available.
25 Alternatively, the look-up and retrieval process 94 can return a probability estimate of availability of a seat conforming to the parameters of the query. In addition, the predictions can return a true/false indication or a probability on a booking code basis.

In addition to being populated with direct queries made
30 by the availability predictor or queries that are obtained from other sources, additional types of query messages can also be used to populate the query database 70. For example, in many

countries it is common for airlines to send out so-called "AVS" (available seat) messages which are distributed from certain airlines, particularly in foreign countries, to other airlines or computer reservation systems. AVS messages specify for a given flight segment whether there are seats remaining on that flight. Sometimes those messages can specify seating on a per booking code basis. Not all airlines use the AVS message process and, therefore, its use in the database would be limited to the availability of such messages for any particular flight segment and airline. Nevertheless, this could be an additional mechanism used to improve the data that is stored in the query database.

Referring now to FIG. 8, a table predictor 65c is shown. The table predictor 65c can be in the form of a three-dimensional table that is stored in computer memory. This is only an example. The table does not have to be three dimensional, and the axes could be different features of an availability query. The table can be indexed by any number of the features of the query. In this example, the table can correspond to the following: the X axis can be a time axis specifying days or hours before departure, the Y axis can be airlines and the Z axis can be booking codes.

The table 150 could have 0's or 1's entries corresponding to not available/available. Alternatively, these entries could also be probability estimates (not shown). This table 150 could be populated by historical information about how often booking codes were available in the past for the airline/booking-code/days-before-departure. For example, if over the past few months availability queries for AA flight 66 that were sent 3 days in advance of travel had booking code Q available 80% of the time, then the probability 0.8 could be stored in the table. When using the predictor 65b, 0.8 could be returned for AA/3day/Q queries, or if an available/not-available

answer was desired, "available" could be answered because $0.8 > 0.5$. In addition, the table could also be populated with the number of seats 154 that might be available on a booking code basis. This can be determined from historical information. The
5 table predictor may also store a number that corresponds to the number of actual queries that were used to arrive at the probability estimate. This number can be used to produce a confidence factor that is returned with the predictor.

Referring now to FIG. 9, a model-based predictor
10 embodiment 65c of the availability predictor 65 is shown. The model-based availability predictor 65c receives 122 a query from a user. The query 163, as shown in FIG. 9A, includes information including an airline 163a, a flight number 163b, a date 163c, an origin and destination (or city pair) 163d, as well as, one or more booking codes 163e. In addition, the query 163 may include other information including point of sale, sales agent, possibly multiple flight numbers, possibly a trip origin and trip destination (as opposed to just the origin/destination of the flights being queried). The query 163 is parsed and analyzed
15 20 by the model-based availability predictor 65c to find features or characteristics of the query 163. That is, the query 163 is broken down to features such as flight number type, period of flight, origin and destination types, the length of time before the flight departs, travel times in the query, and so forth. In
25 addition, the aircraft and capacity, as well as, external events such as sales and strikes, historical availability, and traffic on other flights properties of the traveler and so forth.

For a sample query 163 "UA 100 25JUL98 BOS-CHI Q", the availability predictor can parse 164 that into the following
30 information: the query 163 is for a United Airlines flight, a major carrier, having a flight number 100, a "low number flight", that the date of the flight is in "the summer", and that the

flight is between "two major" cities. The query can determine that the requested fare is a "low cost" fare for Q booking code. In addition, although not present in this query, other factors could also be present in a typical availability query. For example, if the booking code was Y that would indicate a high cost fare. If the flight number is 7500, that could indicate a high flight number and if the origin and destination were "DLH-HIB" (Duluth to Hibbing), that could indicate a flight between two small cities.

Among the features that the availability predictor 65c may take into account are entries in a database of recent or historical fares such as database 70 (FIG. 3). Two features of a query may be "is there a query in the database 70 (FIG. 3) for a similar flight on the same day where the booking code is available" or "is there a query in the database for the same flight on the same day where the booking code is available." The answers in the database 70 (FIG. 3) may be too old to return as an answer, but the information may still be useful in the statistical predictor 65c. This is noted in FIGS. 6 and 10 by the paths between the database and the predictor.

The availability predictor 65c applies the positive, that is, present features of the query to a model and the model returns a prediction of availability corresponding to the query. The results that could be returned from the query may be, for example, a simple "yes", "no", i.e., 1,0 binary return, which indicates either a seat is available or not available or, alternatively, the model may return a number which is or can represent a probability that a seat is available or not available.

Referring now to FIG. 10A, one embodiment 65c' of the model-based availability predictor 65c is shown. The predictor

65c' determines 172 positive features of the query. The predictor 65c' retrieves 174 weights for the positive features with the weights either set in accordance with expert understanding of airline's availability, or, automatically from historical data. In this case, algorithms for setting the weights can be found in various statistics and "machine learning" textbooks such as "Neural Networks for Pattern Recognition" by Christopher Bishop, Oxford Press.

One such algorithm is called "gradient descent" and is approximately as follows:

1. For each feature F, set its weight $W(F)$ to 1.
2. Calculate for each feature F the number of historical queries that returned "available" that the feature occurred in, and call it $H(F)$. (For example, if an American Airlines feature (AA) occurred in 200 queries that were available, then let $H(AA) = 200$).
3. Using the current weights, calculate for each historical query H the probability $P(H)$ of it being available, using the same equations used for normally predicting availability: $P(H) = \exp(X)/(1+\exp(X))$ where $X = \sum W(F)$ for all features F in H.
4. For each feature F, calculate the number of times the current model predicts it will occur in available queries, $M(F)$, by summing $P(H)$ over each historical query H that includes the feature F.
5. Calculate for each feature F the difference between the known number of times the feature

appeared in historical queries, $H(F)$, and the predicted number, $M(F)$, and if for each feature the difference is less than a threshold, stop training and use the current weights.

5 6. Otherwise, update each feature F 's weight $W(F)$ using the formula $W(F) \leftarrow W(F) + K*(H(F) - M(F))$ where K is some small constant like .01.

7. Go to 3 until all weights have been determined.

10 The availability predictor 65c' assigns the weights to the positive factors and adds 176 them to produce a total weight number. The total weight is converted 178 to a total probability. One technique to convert the weight sum to a total probability uses an exponential model of the form $e^x/(1+e^x)$, where x is the total weight number. Alternative models include a
15 linear or quadratic discriminator, factorial model, decision tree, decision list, neural network, sigmoidal network, Bayesian network, naive Bayesian network, Markov random field, maximum entropy model, exponential or log linear model, nearest neighbor model, radial basis model or support vector model and so forth.
20 All of these other models assume that there are features, but not necessarily that the features have weights that are summed.

25 Referring now to FIG. 10B, an alternative embodiment 65c" of the model-based availability predictor 65c (FIG. 8) takes 182 features of the query and uses 184 a decision tree to evaluate the query. From the decision tree, a predicted answer to the query is returned 186. A decision tree is another type of classifier, already listed above. As weights are computed ahead of time in the exponential model the "decision tree" is built ahead of time from historical data. The decision tree is used to

